

⑨ 日本国特許庁 (JP)

⑩ 特許出願公開

⑫ 公開特許公報 (A)

昭55-47549

⑪ Int. Cl.³

G 06 F 15/00

1/04

識別記号

庁内整理番号

7165-5B

6503-5B

⑬ 公開 昭和55年(1980)4月4日

発明の数 1

審査請求 未請求

(全 8 頁)

⑭ 低電力消費マイクロコンピュータ

⑯ 特 願 昭53-119769

⑰ 出 願 昭53(1978)9月28日

⑱ 発 明 者 岩本永三郎

東京都港区北青山3丁目5番17

号

⑲ 出 願 人 日本テキサス・インスツルメン
ツ株式会社東京都港区南青山2丁目24番15
号

⑳ 代 理 人 弁理士 浅村皓 外4名

明 細 書

1 発明の名称

低電力消費マイクロコンピュータ

2 特許請求の範囲

(1) 相補型絶縁ゲート FET で構成されたダイナミック方式の読み取り専用メモリ回路と、ランダムアクセスメモリ回路と、演算回路と、制御回路と、システムクロックを上記回路に供給するクロック発生回路とを包含するマイクロコンピュータにおいて、上記回路の各々の情報非破壊期間がある特定の時期にかかるとし、上記クロック発生回路にホルト入力端子を接続し、上記ホルト入力端子からのホルト信号にตอบสนองして上記クロック発生回路は上記システムクロックを上記特定の時期でのレベルに固定することを特徴とした上記のマイクロコンピュータ。

(2) 特許請求の範囲第1項記載のマイクロコンピュータにおいて、上記回路はプリチャージ期間を上記情報非破壊期間とするダイナミッククロックを含むことを特徴とした上記のマイクロコンピ

ュータ。

(3) 特許請求の範囲第1項ないし第2項記載のマイクロコンピュータにおいて、上記回路は上記特定の時期に情報がスタティッククロックに入っているランダムクロックを含むことを特徴とした上記のマイクロコンピュータ。

(4) 特許請求の範囲第1項ないし第3項記載のマイクロコンピュータにおいて、さらに上記ホルト入力端子と上記クロック発生回路とに接続され、上記特定の時期以前の特定の期間のみ上記ホルト信号を上記クロック発生回路へ通ずるホルト信号ゲート回路を包含することを特徴とした上記のマイクロコンピュータ。

3 発明の詳細な説明

本発明は相補型絶縁ゲートトランジスタを用いた大規模集積回路 (LSI) マイクロコンピュータに関するものであり、特に半導体チップ面積が小さく、且つ消費電力を低く抑えることのできるマイクロコンピュータを提供するものである。

集積回路 (IC) における絶縁回路はその機構の

上から通常スタティック方式とダイナミック方式とに大別される。

スタティック方式はフリップ・フロップ回路、あるいはラッチ等のスタティックロジックのみを用いたものであり、情報の保持に時間的制限がないという点で有利である反面、素子数が多くなる、従つて、半導体チップの面積も大きくなるという不利な点がある。

一方、ダイナミック方式は絶縁ゲート型トランジスタのゲート浮遊容量に蓄積された電荷を情報の保持に利用するダイナミックロジックを含んだものであり、同一の回路で比較した場合、スタティック方式より素子数が少なくなる、云いかえれば同一面積の半導体チップに大量の論理回路を作れるという利点がある。その反面、ゲート浮遊容量の電荷はリーク電流によつて消滅していくので一定時間以内(数ミリ秒以内)の周期で再書き込み(リフレッシュ)を繰り返さねばならず、リフレッシュが必要なことにより消費電流が大きくなるという不利な点がある。

特に最近では例えば電池で駆動できるように低電力消費のもので、ワンチップマイクロコンピュータで代表されるような小面積の半導体チップに大量の論理回路を組み込んだものが要望されている。

低電力消費の要望を満たす方法として、従来の n チャネルや p チャネル絶縁ゲートトランジスタの代りに相補型絶縁ゲートトランジスタ(これにはいくつかの種類があるが本明細書ではこれらを総称して「 $0-MOS$ 」と呼ぶことにする)を用いることが考えられた。しかし、 $0-MOS$ により確かに電力消費を減少させ得るが、スタティック方式を採用した場合素子数が多くなり半導体チップ面積が増大してしまふ。またダイナミック方式を採用しても、実行速度を上げるため高周波数のクロック信号で操作しようとする、 $P=0V^2f$ (但し、 P :電力、 0 :容量、 V :電圧、 f :周波数)の関係から周波数 f の上昇による電力消費は増加し、 $0-MOS$ を用いたことによる利点があり顕著とならなくなる。

ここでダイナミック方式について詳しく検討し

てみると、通常完全ダイナミックロジックで作られるROMやRAMにはプリチャージ期間が必要であり、このプリチャージ期間に内部システムクロックを止めたとしても情報の破壊が起らないことがわかる。つまり完全ダイナミックロジックには情報非破壊期間と呼べる期間が存在する。またダイナミック方式のランダムロジックにおいても上述のラッチのようなスタティックロジックの部分を含んでおり、必要な情報がそのようなスタティックロジック内にあるか、またはダイナミックロジックの入力がそのようなスタティックロジックに直接接続されている期間にシステムクロックを止めても情報の破壊が起らないことがわかる。つまりランダムロジックにおいても情報非破壊期間と呼べる期間が存在する。

本発明は上述の点に着目してなされたもので、従来のダイナミック回路部分のそのような情報が破壊されないタイミングがその回路部分の都合だけを考へて各自バラバラに設定されていたのを改め、ダイナミック方式においてある特定のタイミ

ング期間中は、必要とする情報が入っている回路部分が全て上述の情報の破壊の起らない状態になるようにシステムのタイミング設計を行い、その特定のタイミング時点でシステム全体にわたつてシステムクロックを止めることにより、ダイナミック方式の小面積大容量という利点と $0-MOS$ の低電力消費という利点との両方を有するマイクロコンピュータを提供することを目的とする。

本発明の他の目的はこのマイクロコンピュータの入力端子のひとつにここでハルト(HALT)信号と呼ぶレベル信号を加えることにより、コンピュータ内部の全てのクロックをあるタイミングで外部から停止させることができるマイクロコンピュータを提供することである。

従来のハルトあるいはホールド(HOLD)端子と呼ばれるものはそのハルトあるいはホールド信号に反応してコンピュータのソフトウェアによる指令で演算を停止させることを意味していた。その場合演算がなされない、いわゆるアイドル(IDLE)状態を保っているだけで、ダイナミックメモリは

依然として駆動しておりリフレクシュされていた。従つて、従来のホルト時の電力消費は他の状態時のそれと何ら異なるものではなかつた。本発明でいうホルトあるいはホールド(以下、これらを代表して「ホルト」と呼ぶ)とはダイナミックロジックの駆動に使用されるシステムクロックを完全に停止させることにより、消費電力をリーク電流のオーダーにまで下げてしまうものである。これにより従来はホルト状態においてさえダイナミックロジックに必要な15ミリワット(mW)の消費電力を、本発明によれば高々5マイクロワット(μ W)のオーダーにまで下げることができるのである。

またソフトウェアの命令セットの内に、本発明のホルト状態にする命令を組みこむことも考えられるが、この場合はソフトウェアの処理に必要な時間だけ遅れが生じることを留意しなければならぬ。本発明のように外部端子からハードウェアでホルト指令を行えば信号処理のスピード、容易さにおいて利点が生じる。

本発明の他の目的はホルト状態にするタイミング

より前の特定の期間でのみホルト信号を受け付けることにより誤動作を防止したマイクロコンピュータを提供するものである。

以下本発明の実施例を図面に従い具体的に説明する。

第1図は本発明の実施例であるマイクロコンピュータ全体を示すブロック図である。図に於いてマイクロコンピュータ1は、単一の半導体チップ上に脱取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、演算回路、制御回路等を相補導電型ゲートトランジスタ(CMOS FET)集積回路により構成したものを含んでいる。このマイクロコンピュータは、使用者が作成するプログラムに従つて、ゲート用マスクを製造過程で変更し、チップ内部の固定ROMをコーディングするものである。このマイクロコンピュータは、単一の半導体基板上に構成することができる。マイクロコンピュータ1の構成要素のうち主なものの動作を以下に説明する。

ROM の動作

マイクロコンピュータ1の中に組込まれたROM 2は、1024ワード×8ビットの命令を保持可能であつて、この命令によりマイクロコンピュータ1内の素子の動作を決定する。ROM 2は16ページに分けられ、各ページに64の命令が入る。電源の投入によりプログラムはある決つた番地からスタートし、その後、シフトレジスタのプログラムカウンタPCが各ページのROM命令を順次実行していく。条件付ジャンプ命令またはコールサブルーチン命令は、8ビットのプログラムカウンタアドレスをソフトウェア制御へ移すよう変更し、各サブルーチンリターンアドレスはサブルーチンリターンレジスタに記憶される。ページアドレスレジスタ(4ビット)PAは、16個のROMページのうち現在実行中のページを記憶している。

RAM の動作

RAM 3は、アドレス指定可能な256ビットからなり、16ワード×4ビットの4つのファイルで構成されている。RAM 3はレジスタXREGとレジスタYREGによつてアドレスされるかイレ

ジスタは演算回路4によつて制御され、1ファイル中の16ワードのうちの1ワードを指定する。

このマイクロコンピュータ1に於いては、「Yの内容を定数と比較」、「Xに定数をセット」、「Yの内容を1増やす」、「Yの内容を1減らす」、「Yから、またはYへデータの転送を行う」などの命令がある。

レジスタXREGの中の2ビットは、RAM 3の4つのファイルの1つを選択する。レジスタXREGには定数がかけられるか、または補数がとられ、4ビットのデータワードはアキュムレータACCまたはROM 2の定数によつて決まるレジスタXREG、レジスタYREGによつて指定されるRAM番地(location)へ転送される。RAM 3からの出力ワードは演算回路4で処理され、1命令間隔を置いてレジスタYREGまたはアキュムレータACCに転送される。RAM 3内の全てのビットは、セット/リセットまたはビットテストができる。

演算回路ユニット(ALU)の動作

演算および論理演算動作は4ビットの加算器と、それに付随する論理回路によつて実行され、演算回路は論理演算比較、算術演算比較、加減算を行う。

入出力

マイクロコンピュータ1には、 E_1, E_2, E_4, E_8 および L_1, L_2, L_3, L_4 の8個のデータ入力があり、これらは多重化されて4ビットの入力バスへと導入される。その他にこのマイクロコンピュータ1は E/L セレクタとモードセレクタという制御入力を持つている。 E/L 制御入力は内部のプルダウンレジスタを持ち、入力がない場合又は、 E/L 入力が低レベルの場合、 E 入力が選択される。また E/L 入力が高レベルにある場合は、4ビットの L 入力が選択される。

このマイクロコンピュータ1には、多目的の用途に適するよう、 R 出力と O 出力の2つの出力チャンネルがある。 R 出力は通常は入力をマルチプレクサするものであるが、表示出力、外部メモリ出力、または外部装置への出力データとしての O

出力に同期をかけるためにも使われる。

インストラクションPLA (プログラマブルロジックアレイ)

プログラマブルな命令は、インストラクションPLA 8により定義される。32個のプログラマブル入力NANDゲートは8ビットの命令語をデコードする。各々のNANDゲートは出力は16個のマイクロインストラクションの組合せを選択する。この16個のマイクロインストラクションは演算回路(ALU)、ステータスラッチ、およびRAM 3の書き込み入力を制御する。

クロック発生回路

マイクロコンピュータ1内部の信号処理は端子0801, 0802からの外部クロックをもとにクロック発生回路5が作るシステムクロック(本実施例の場合 ϕ_x, ϕ_y, ϕ_z)に同期して行なわれる。本発明のひとつの特徴は、このクロック発生回路5に ϕ 対し第1図に示すHALTで示すホルト信号を入力してクロック発生回路5からのシステムクロックを停止することである。この点については

後で詳細に説明する。

タイミング

本発明の実施例において、ひとつの命令サイクルは4クロックサイクルからなり、すべての命令はひとつの命令サイクル内で実行される。実際のマシンサイクルタイムは、オシレータ0801とオシレータ0802のピンに接続される抵抗およびコンデンサ、あるいは0801に加える外部クロック入力周波数のいずれかによつて決定される。一例として、命令サイクルを6~120マイクロ秒とし、各クロック・サイクルをその1/6の1~20マイクロ秒とすることができる。

第2図に第1図図示の本発明実施例における出力、入力及び命令のタイミングチャートを右方向に時間の流れをとつて示す。左半分に示す第1の命令サイクルは、 N 番目の命令を取出す段階(fetch cycle)であり、同時にそのひとつ前の($N-1$)番目の命令を実行する段階(execute cycle)である。右半分に示す第2の命令サイクルは、上記第1命令サイクルで取出した N 番目の

命令を実行する段階であり、同時に次の($N+1$)番目の命令を取出す段階である。図の下半分には、 N 番目の命令に関する種々の操作のタイミングが、クロックサイクル T_1 ないし T_6 に対応して記述されている。

N 番目の命令に関して言えば、ROMのアドレスは第1命令サイクルの T_1 から T_3 の間になされ、 T_4 で取出され、 T_5 においてはプログラムカウンタが更新され、プランタ・コールが実行される。第2命令サイクルにおいて、インストラクションは $T_1 \sim T_6$ にわたつて実行され、 $T_1 \sim T_3$ でRAMの読み出しを行い、 T_4 でRAMに書き込み、 T_5 と T_6 でALUに

本実施例の場合ROM 2、RAM 3、インストラクションPLA 8等は完全ダイナミッククロックで作られている。完全ダイナミッククロックの命令サイクルでの状態を示す例としてROM、RAMについてのみ第2図に示してある。第2図から明らかなように完全ダイナミッククロックであるROM、RAMのプリチャージの期間の少なくとも一部が T_6 の

タイミングにかかっているようにシステム設計されている。またその他のダイナミック方式のランダムロジックについても第2図で例として示したALUの波形から明らかのようにホルト時に情報の保持が必要なものも少なくともT6のタイミングのときにはその情報がスタティックロジック(つまりその情報がゲート浮遊容量で保持されていない状態)に入るようにタイミング設計されている。第2図のROM, RAM, ランダムロジックの波形で太線で示した部分は上述のプリチャージ時、および情報保持可能時を示している。

ホルト信号の動作

第3図は、第1図図示の本発明実施例におけるクロック発生回路5を示す。この回路は、大規模集積回路(LSI)の一部として、また第1図のマイクロコンピュータの一部として、半導体基体上に形成されており、発振入力端子OSC1, OSC2, ホルト信号入力端子HALT; 発振回路54主としてトランスファゲートで構成された6進カウンタ55; およびクロック出力回路56とを含んでい

15

言い換えれば、システム中の信号の遅れに対する考慮に余裕を与え、設計を容易なものにすることができるのである。

第3図において、発振回路部54に対し、外部の発振器から、発振入力端子OSC1, OSC2を通じて、第4図(a)及び(b)に示される信号がそれぞれ供給される。ホルト信号の入力端子53に低レベルの信号(たとえば V_{SS} のレベル)が入っていると、このマイクロコンピュータは通常の動作を継続して行い、逆に高レベル(たとえば V_{DD} のレベル)の信号が入ると、ホルト機能が働き、コンピュータ全体のシステムが停止する。

まずホルト信号入力端子HALTに低レベル信号が与えられている場合、即ちホルト機能が動作していない場合、第4図(a), (b)に示される発振信号は、発振回路54に入り、6進カウンタ55及びクロック出力回路56を通過して第4図(a), (b)に示されるシステムクロック ϕ_X, ϕ_Y および ϕ_Z が出力される。本発明においては、ひとつの命令サイクルを構成するT1からT6のタイミングに

17

特開昭55-47549(5)
る。ここでクロック出力回路56から出る出力信号 ϕ_X, ϕ_Y, ϕ_Z は第1図のマイクロコンピュータのほとんど全ての回路に供給されており、それらを動作する基本的なシステムクロックである。第1図では、実質に従い、この出力回路56とこれらシステムクロックの供給先の配線は省略している。

本発明の実施例では、第3図のホルト信号ゲート回路7により高レベルのホルト信号が実行サイクル(execute cycle)のT2からT4までの間のみ受けるようにしている。すなわち、T5, T6でホルト信号を発生しても次のサイクルのT2になつてはじめてホルトがアクセプトされる。このように次第にそのホルト信号が実行されるT6(実行段階の最終タイミング)との間に、T5という時間をあけ、ホルト信号の入力を遅延させるようにしている。このように実行段階の最終タイミングのひとつ前のタイミングをホルト信号の入力タイミングとして使わないことによつて、システム中の信号の遅れに原因する不確かさを排除できる。

16

対応して、6種のシステム・クロックを使わずに、その半分の3種のシステム・クロック ϕ_X, ϕ_Y, ϕ_Z を使用し、各回路を動作させる基本的なクロック信号とすることにより、クロックサイクルがT1からT6のいずれのタイミングにあるかはシステムクロック ϕ_X, ϕ_Y, ϕ_Z の組合せから求められる。

この状態で高レベルのホルト信号が第3図の入力端子HALTに入ると、発振回路54の出力動作が停止する。そのとき、タイミングT1~T6に無関係にホルト機能が動作すると、情報の蓄積される位置が特定できない等の欠点を生ずる。そこで本発明は、ひとつの命令サイクルT1~T6のうち予め定められた任意のタイミング(本実施例の場合T2~T4)のときだけホルト信号を受けつけるように、ホルト信号ゲート回路7がホルト信号と他のタイミング信号(本実施例の場合 ϕ_Z)とを逐次づけWOR回路57に入力する。この実施例では、前述のように命令サイクルの最終タイミングであるT6になつてからホルト機能が働くように、

18

T6に相当するタイミングに信号が6進カウンタ58からNOR回路57に入る。本実施例の場合、NOR回路57には $\overline{\text{HALT}}$ 、 $\overline{\text{P}_X}$ 、 $\overline{\text{P}_Y}$ が入力され、 $\overline{\text{P}_X}$ 、 $\overline{\text{P}_Y}$ で命令タイトル中の特定のタイミングT6を決定し、 $\overline{\text{HALT}}$ 、 $\overline{\text{P}_X}$ 、 $\overline{\text{P}_Y}$ 、 $\overline{\text{P}_Z}$ が全て'0'のときハルト機能を実行させている。T2-T4でのみハルト信号を受けつけるのでシステムの停止が安定して行われる。

このハルト動作が実行される特定のタイミングT6のときはダイナミック回路内での情報は必ずスタティック・ロジックに入っているようにシステムが組まれているので、システムクロックが停止しても必要な情報が保持される。このようにして、ダイナミック・ロジックの利点である小面積高密度の集積度を稼ぐことができると同時に、ハルト機能により補償遅延ゲートFETで構成された回路内に蓄積された情報は破壊されることなく保持でき且つ保持している回路の消費電力が極めて低いレベルにまで減少するのである。

なお本明細書中において0-MOSなる用語を用い

ているが、本発明はメタルゲートのFETに限られるものではなく、メタルゲートでもシリコンゲートでもあるいは他のゲート層のFETでも適用可能なことは明らかである。さらに本明細書中の相補型または0-MOSなる用語は広い意味として用いられているのであり、例えばROM等に採用されているPチャンネルMOSとNチャンネルMOSとが夫々分離して作り込まれており、回路全体で見るとPチャンネルMOSとNチャンネルMOSとの両方が使われているものをも指している点に注意すべきである。

4. 図面の簡単な説明

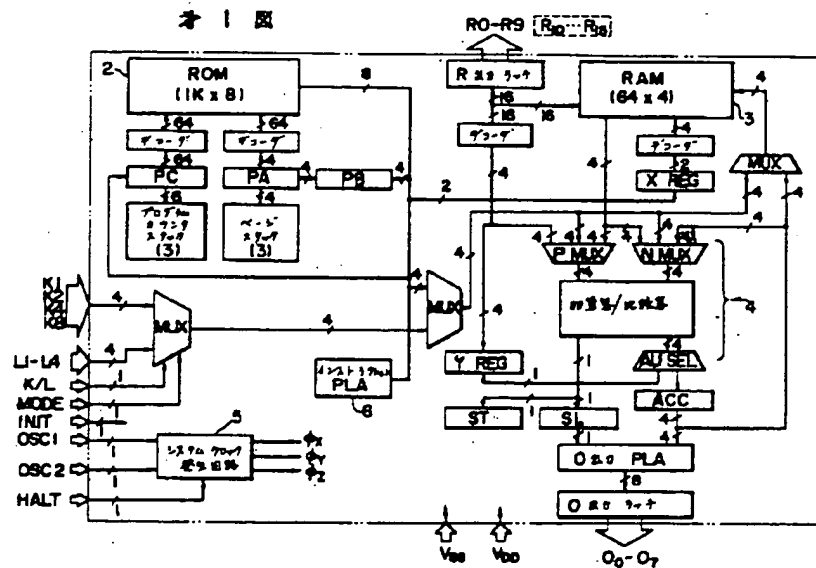
第1図は本発明のマイクロコンピュータの一実施例を示すブロック図である。

第2図は第1図図示の実施例の動作状態を示すタイミング図である。

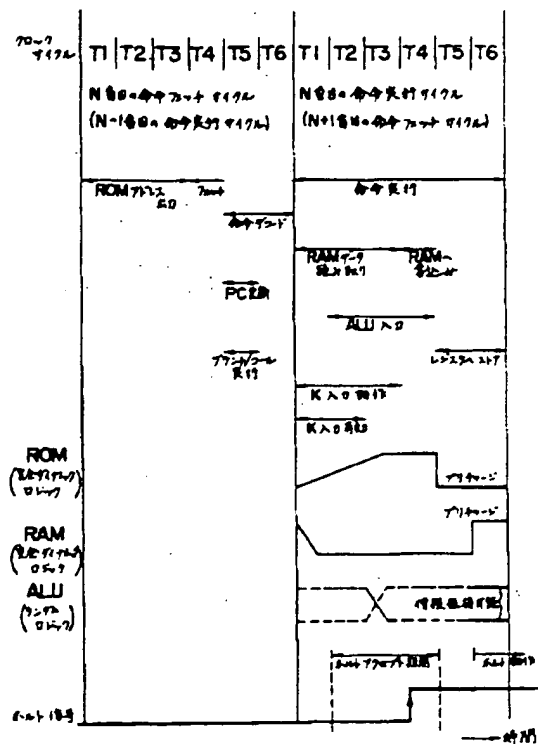
第3図は第1図図示の実施例の主要回路の電気回路図である。

第4図は第3図図示の回路の動作状態を示すタイミング図である。

- 1…マイクロコンピュータ、2…ROM、3…RAM、
4…復算回路、5…クロック発生回路、
6…インストラクションPLA、7…ハルト信号ゲート回路。



才 2 図



才 3 図

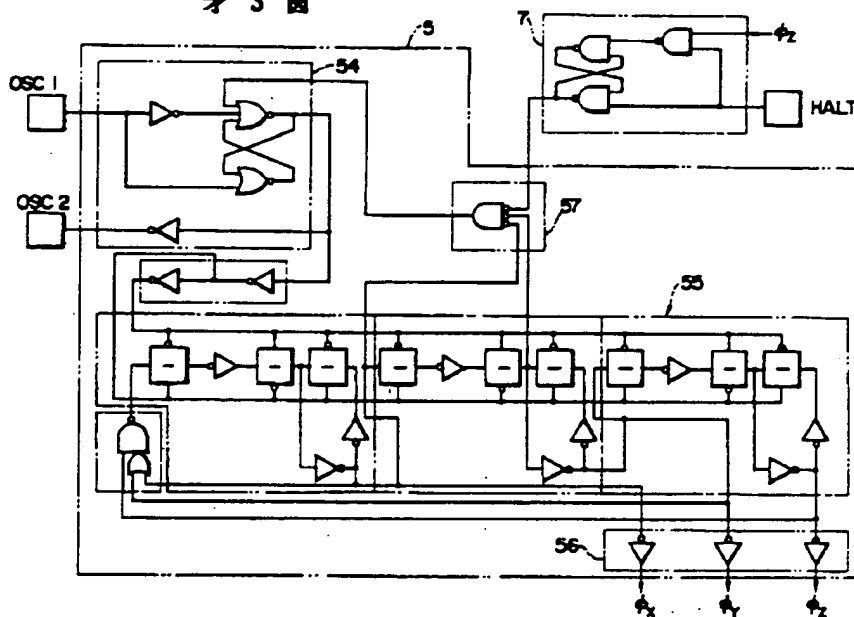
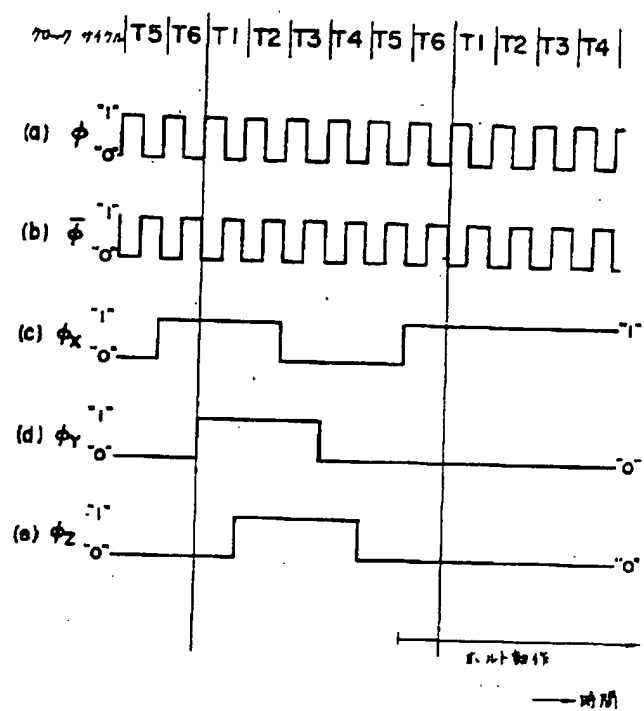


图 4



60 10.19

特許法第17条の2の規定による補正の掲載

昭和53年特許願第 119769 号(特開 昭 55-47549 号, 昭和55年 4 月 4 日 発行 公開特許公報 55-476 号掲載)については特許法第17条の2の規定による補正があったので下記のとおり掲載する。 6 (3)

Int. Cl. 1	識別記号	庁内整理番号
G06F 15/00 1/04		8549-5B 7157-5B

手続補正書

昭和60年 7 月 29 日

特許庁長官殿

1. 事件の表示

昭和53年特許願第 119769 号

2. 発明の名称

低電力消費マイクロコンピュータ

3. 補正をする者

事件との関係 特許出願人

方式

審査

住所

氏名

(名称)

日本デヤサス・インスツルメンツ株式会社

4. 代理人

住所

氏名

〒100 東京都千代田区大手町二丁目2番1号

新大平町ビルディング331

電話 (211) 3 6 5 1 (代表)

(8889) 浅 村 昭

5. 補正命令の日付

昭和 年 月 日

6. 補正により増加する発明の数

7. 補正の対象

明細書の特許請求の範囲

8. 補正の内容 別紙のとおり

9. 添付書類の目録 同時に出願審査請求書を提出してあります。

特許請求の範囲を別紙の通り訂正する。

「2. 特許請求の範囲

- (1) 相補型絶縁ゲート FET で構成されたダイナミック方式の読み取り専用メモリ回路と、
データアクセスメモリ回路と、演算回路と、制御回路と、システムクロックを上記回路に供給するクロック発生回路とを包含するマイクロコンピュータにおいて、上記回路の各々の情報非破壊期間がある特定の時期にかかるようにし、上記クロック発生回路にホルト入力端子を接続し、上記ホルト入力端子からのホルト信号に応答して上記クロック発生回路は上記システムクロックを上記特定の時期でのレベルに固定することを特徴とした上記のマイクロコンピュータ。
- (2) 特許請求の範囲第1項記載のマイクロコンピュータにおいて、上記回路はプリチャージ期間を上記情報非破壊期間とするダイナミッククロックを含むことを特徴とした上記のマイクロコンピュータ。
- (3) 特許請求の範囲第1項または第2項記載のマイクロコンピュータにおいて、上記回路は上

記特定の時期に情報がスタティックロジックに入っているランダムロジックを含むことを特徴とした上記のマイクロコンピュータ。

(4) 特許請求の範囲第1項から第3項のいずれかに記載のマイクロコンピュータにおいて、さらに上記ホルト入力端子と上記クロック発生回路とに接続され、上記特定の時期以前の特定の期間のみ上記ホルト信号を上記クロック発生回路へ通すホルト信号ゲート回路を包含することを特徴とした上記のマイクロコンピュータ。」

Japanese Kokai Patent Application No. Sho 55[1980]-47549
[Without claims, as requested]

JAPANESE PATENT OFFICE
PATENT JOURNAL (A)
KOKAI PATENT APPLICATION NO. SHO 55[1980]-47549

Int. Cl. ³ : G 06 F	15/00 1/04
Sequence Nos. for Office Use:	7165-5B 6503-5B
Filing No.:	Sho 53[1978]-119769
Filing Date:	September 28, 1978
Publication Date:	April 4, 1980
No. of Inventions:	1 (Total of 8 pages)
Examination Request:	Not filed

LOW POWER CONSUMPTION MICRO-COMPUTER

Inventor:	Eizaburo Iwamoto, 3-5-17 Kitaaooyama, Minato-ku, Tokyo-to
Applicant:	Nippon Texas Instruments K.K., 2-24-15 Minamiaoyama, Minato-ku, Tokyo-to
Agents:	Hiroshi Asamura, and 4 others

[There are no amendments to the patent.]

* * *

Detailed explanation of the invention

This invention concerns a large-scale integrated circuit (LSI) micro-computer that uses a complementary insulated gate transistor. It particularly offers a micro-computer in which the area of the semiconductor chip is small, and the consumption of power can be reduced.

Logic circuits in an integrated circuit (IC) can generally be classified as using a static method or a dynamic method in the mechanism.

The static method uses only static logic, such as a flip flop circuit or a latch, for example, which is effective in that there is no time limit in the preservation of information, but on the other hand a disadvantage exists in a large number of elements, which accordingly increases the area of a semiconductor chip.

On the other hand, the dynamic method includes dynamic logic, which utilizes electric charges that accumulate in the gate flotation capacity of an insulated gate type transistor in the preservation of information. When comparing the same circuit, it uses a lower number of elements than in the static method. In other words, it has the benefit of enabling the formation of a large quantity of logic circuits in a semiconductor chip in the same area. On the other hand, it has a disadvantage in that rewriting (refreshing) must be repeated at a frequency within a constant period of time (within several milliseconds) because electric charges in the gate flotation capacity decrease from leakage current, and the power consumed increases because refreshing is necessary.

Particularly recently, micro-computers that have a low power consumption and can be operated by batteries, for example, and in which a large number of logic circuits are integrated into a semiconductor chip in a small area, which can be represented by one-chip micro-computers, for example, have been requested.

As a method for satisfying the demand for low power consumption, the use of a complementary type insulated gate transistor (of several types, but they are comprehensively referred to as a "C-MOS" in this specification) has been considered instead of conventional P-channel and N-channel insulated gate transistors. Although the power consumption can certainly be decreased by a C-MOS, the number of elements increases when the static method is used, and the area of a semiconductor chip increases. Also, even if the dynamic method is used, the power consumption due to an increase in the frequency f increases because of the relationship of $P = CV^2f$ (P : power, C : capacity, V : voltage, and f : frequency) when an attempt is made to operate by a high frequency clocking signal for increasing the execution speed, and the advantage of using a C-MOS is not significantly displayed.

The dynamic method will be examined in detail here. ROMs and RAMs that are prepared by completely dynamic logic generally require a pre-charging period, and it can be understood that the destruction of information does not occur even if the internal system clock is stopped during this pre-charging period. More precisely, the completely dynamic logic has a period which can be called an information non-destruction period. The random logic of the dynamic method also includes a portion of static logic like the latch described above, and it can be understood that the destruction of information does not occur when the system clock is stopped when necessary information is within such a static logic or during the period when the input of the dynamic logic is directly connected to such static logic. In other words, a period which can be called the information non-destruction period also exists in the random logic.

The objective of this invention, which was made while considering the point described above, is to improve the individual and independent setting of the timing for not destroying such information in the area of each conventional dynamic circuit while considering only the convenience of that circuitry part, to design the system timing so that the entire circuitry part containing the necessary information is in a state without the occurrence of the

destruction of information described above during the specific timing by the dynamic method, and to offer a micro-computer having both the advantage of a small area and large capacity of the dynamic method and the advantage of a low power consumption of a C-MOS by stopping the system clock throughout the system at that specific timing point.

Another objective of this invention is to offer a micro-computer which can stop all clocks within the computer at a certain timing from the outside by adding a level signal referred to herein as a halt (HALT) signal into one of the input terminals of this micro-computer.

The conventional type referred to as a halt or hold (HOLD) terminal meant allowing the operation to rest by a command of the software of a computer in response to said halt or hold signal. In that case, the dynamic memory was still operating and was refreshed even when not operating, in other words, while maintaining a so-called idle (IDLE) state. Accordingly, the power consumption during a conventional halt period was not different from other states. The halt or hold referred to in this invention (they will be representatively referred to as "halt" below) decreases the consumption power to the order of a leakage current by completely stopping the system clock, which is used in the operation of the dynamic logic. Through this, the conventional consumption power of 15 milliwatts (mW) necessary for the dynamic logic even in the halt state can be lowered to the order of 5 microwatts (μ W) at most in this invention.

The integration of a command for attaining the halt state in this invention into the set of commands in the software can also be considered, but in this case, one must consider the generation of a delay for the time necessary for the processing of the software. The execution of the halt command by hardware through an external terminal in this invention has the advantage of fast and simple signal processing.

Another objective of this invention is to offer a micro-computer in which erroneous operations are prevented by receiving a halt signal only during a specific period prior to the timing for obtaining the halt state.

An application example of this invention will be concretely explained in accordance with the figures below.

Figure 1 is a block diagram which shows an entire micro-computer as an application example of this invention. In the diagram, a micro-computer (1) includes a read-only memory (ROM), random access memory (RAM), arithmetic circuit, and a controlling circuit, etc. over a single semiconductor chip constructed by a complementary type insulated gate transistor (C-MOSFET) integrated circuit. This micro-computer changes the gate mask in the manufacturing process according to a program prepared by the user, and coats the fixed ROM within the chip. This micro-computer can be constructed over a single semiconductor substrate. Operations of the main structural elements of the micro-computer (1) will be explained below.

Operation of ROM

A ROM (2) integrated into the micro-computer (1) can hold commands of 1024 words x 8 bits, and the operation of the elements within the micro-computer (1) are determined through these commands. The ROM (2) is sectioned into 16 pages, and each page can contain 64 commands. The program starts from one determined address by turning on the power source, and the program counter PC of the shift register afterwards executes the ROM

commands in each page successively. A branch command with conditions attached or call sub-routine command changes to move the 6-bit program counter address to software control, and each sub-routine return address is stored in the sub-routine return register. Page address register (4 bits) PA stores the page executing at present out of the 16 ROM pages.

Operation of RAM

A RAM (3) consists of 256 bits which can specify an address, and is comprised of 4 files of 16 words x 4 bits. The RAM (3) is addressed either by the X register XREG or the Y register YREG, or the Y register is controlled by the arithmetic circuit (4), and 1 word out of 16 words in 1 file is specified.

This micro-computer (1) includes commands like “comparing the contents of Y to a constant,” “setting Y to a constant,” “increasing the content of Y by 1,” “decreasing the content of Y by 1,” and “transferring data from Y or to Y,” for example.

2 bits in the X register XREG select 1 of the 4 files in the RAM (3). Either a constant is placed in the X register XREG, or a complement is used, and a data word in 4 bits is transferred to an accumulator ACC, X register XREG determined by the constant of the ROM (2), or the RAM address (location) specified by the Y register YREG. The output word from the RAM (3) is processed at the arithmetic circuit (4), and transferred to the Y register YREG or the accumulator ACC after an interval of 1 command. All of the bits within the RAM (3) can be set/reset, or bits can be tested.

Operation of the arithmetic circuit unit (ALU)

The arithmetic and logic arithmetic operations are executed by a 4-bit adder and a logic circuit, which is attached to it, and the arithmetic circuit implements logic arithmetic comparisons, arithmetic calculation comparisons, and additions and subtractions.

Input and output

The micro-computer (1) has 8 data inputs, which are K1, K2, K4, and K8 and L1, L2, L3, and L4, and they are multiplexed and introduced into a 4-bit input bus. In addition to this, micro-computer (1) has a control input referred to as a K/L selector and a mode selector. The K/L control input has an internal pull-down register, and selects input K when there is no input or when the K/L input is at a low level. The 4-bit input L is selected when the K/L input is at a high level.

Micro-computer (1) has 2 output channels, which are output R and output 0, to suit multi-purpose applications. Output R generally multiplexes inputs, but it is also used for synchronizing the display output, external memory output, and output 0 as the output data to an external system.

Instruction PLA (programmable logic array)

Programmable commands are defined by the instruction PLA (6). 32 programmable input NAND gates decode command words in 8 bits. Each NAND gate selects combinations of 16 micro-instructions. These 16 micro-instructions control the write-in input to the arithmetic circuit (ALU), status latch, and the RAM (3).

Clock generating circuit

Signal processing within the micro-computer (1) is executed based on an external clock from terminals OSC1 and OSC2 and in synchronization with the system clock (ϕ_X , ϕ_Y , and ϕ_Z in this application example) formed by the clock generating circuit (5). One of the characteristics of this invention is to input a halt signal indicated by HALT in Figure 1 to this clock generating circuit (5) and stop the system clock of the clock generating circuit (5). This point will be explained in detail later.

Timing

One command cycle in the application example in this invention consists of 6 clock cycles, and all of the commands are executed within one command cycle. Actual machine cycle time is determined by either the oscillator OSC1, a resistance and a capacitor that are connected to pin of the oscillator OSC2, or an external clock input frequency added to OSC1. As one example, when the command cycle is 6-120 micro-seconds, each clock cycle of 1-20 micro-seconds, which is 1/6 of said cycle, can be used.

Figure 2 shows timing charts of outputs, inputs, and commands in the application example of this invention illustrated in Figure 1 with the flow of time in the direction to the right. The first command cycle indicated in the left side half is a stage for fetching the Nth command (fetch cycle), which is also a stage for executing the preceding command (N-1) (execute cycle) at the same time. The second command cycle indicated in the right half is a stage for executing the Nth command, which is fetched in the aforementioned first command cycle, which is also a stage for fetching the next (N+1) command at the same time. Timings of various operations with respect to the Nth command are described corresponding to the clock cycle T1 through T6 in the lower half of the figure.

With respect to the Nth command, the ROM is addressed between T1 and T3 of the first command cycle, fetched at T4, and the program counter is renewed at T5, and a branch/call is executed. In the second command cycle, the instruction is executed over T1-T6, the RAM is read out at T1-T3, the RAM is written at T4, and the ALU is input at T3 and T4.

In this application example, the ROM (2), RAM (3), and the instruction PLA (6), etc. are formed of completely dynamic logic. Figure 2 shows only the ROM and the RAM as an example indicating a command cycle of completely dynamic logic. As is clearly indicated in Figure 2, the system is designed so that at least one portion of the pre-charging period of the ROM and the RAM that have completely dynamic logic falls over the timing at T6. With other dynamic method random logics as well, as clearly shown in the waveform of ALU indicated as an example in Figure 2, the timing is designed so that information that requires the preservation of the information during the halting time enters the static logic (in other words, a state exists in which the information is not held in

the gate flotation capacity) at least at the timing of T6. Areas indicated by thick lines in the waveforms of ROM, RAM, and the random logic in Figure 2 indicate the pre-charging time described above and the time possible for the preservation of information.

Operation of the halt signal

Figure 3 shows the clock generating circuit (5) in the application example in this invention illustrated in Figure 1. This circuit is formed over a semiconductor substrate as a part of a large scale integrated circuit (LSI) or as a part of a micro-computer in Figure 1, and includes oscillation input terminals OSC1 and OSC2, halt signal input terminal HALT, oscillation circuit (54), a sexenary counter (55) mainly comprised of a transfer gate, and a clock output circuit (56). Output signals ϕ_X , ϕ_Y , and ϕ_Z that are output from the clock output signal (56) here are supplied to almost all of the circuits in the micro-computer in Figure 1, and a basic system clock operates them. Figure 1 is in accordance with a commonly used technique, and the output circuit (56) and the wires with the supplying ends of these system clocks are omitted.

In the application example in this invention, halt signals at a high level are received from the halt signal gate circuit (7) in Figure 3 only during the execution cycle (execute cycle) between T2 and T4. More precisely, even though halt signals are generated at T5 and T6, a halt is accepted for the first time at T2 in the next cycle. In this manner, the period of T5 is assured before T6 for an actual execution of that halt signal (the final timing in the execution stage), and inputting of the halt signal is made reliable. In this manner, the unreliability resulting from a delay in the signal in the system can be eliminated by not using the timing that is one timing before the final timing in the execution stage as the input timing of the halt signal. In other words, a margin is given to the delay of the signal in the system, and a simple design can be attained.

In Figure 3, signals indicated in Figure 4(a) and (b) are respectively supplied to the oscillation circuit (54) from an external oscillator through oscillation input terminals OSC1 and OSC2. If a signal at a low level (level V_{SS} , for example) has entered the input terminal (53) of the halt signal, this micro-computer continuously operates normally. On the other hand, when a signal at a high level (level V_{DD} , for example) enters, the halting function operates, and the entire computer system stops.

First, when a low level signal is given to the halt signal input terminal HALT, in other words, when the halting function is not operating, the oscillation signals indicated in Figure 4(a) and (b) enter the oscillation circuit (54), pass through the sexenary counter (55) and the clock output circuit (56), and system clocks ϕ_X , ϕ_Y , and ϕ_Z indicated in Figure 4(c), (d), and (e) are output. In this invention, instead of using 6 system clocks corresponding to T1-T6 timings which make up one command, 3 system clocks ϕ_X , ϕ_Y , and ϕ_Z , which is half, are used as basic clock signals which operate each of the circuits. A timing of the clock cycle at any of T1 through T6 can be obtained through a combination of the system clocks ϕ_X , ϕ_Y , and ϕ_Z .

When a halt signal at high level enters the input terminal HALT in Figure 3, the output operation of the oscillation circuit (54) stops. During this, a disadvantage occurs in which the position of the storage of the information cannot be specified when the halting function operates regardless of timings T1-T6. Therefore, in this

invention, the halt signal gate circuit (7) inputs [signals] into the NOR circuit (57) while correlating between the halt signal and other timing signals (ϕ_z in this application example) in a manner so that the halt signal is received only at a pre-determined optional timing (T2-T4 in this application example) out of one command cycle T1-T6. In this application example, the signal enters from the sexenary counter (55) into the NOR circuit (57) at a timing equivalent to T6 so that the halting function operates at T6, which is the final timing of the command cycle, as described above. In this application example, $\overline{\text{HALT}}$, ϕ_x , and ϕ_y are input into the NOR circuit (57), specific timing T6 in the command cycle is determined at ϕ_x and ϕ_y , and the halting function is executed only when all of $\overline{\text{HALT}}$, ϕ_x , ϕ_y , and ϕ_z are "0." Stopping of the system can be stably obtained because the halt signal is received only at T2-T4.

The system is integrated so that the information within the dynamic circuit always enters the static logic at specific timing T6 for the execution of this halting operation. Therefore, necessary information is preserved even if the system clock stops. In this manner, high density integration in a small area, which is an advantage of dynamic logic, can be obtained, and at the same time information, which has been stored within a circuit constructed by a complementary type insulated gate FET, can be preserved without being destroyed by the halting function, and the consumption of power during the preservation decreases to a very low level.

The term C-MOS is used in this specification, however, this invention is not limited only to metal gate FETs, and it is clear that application with metal gates, silicon gates, and other gate type FETs is possible. Furthermore, one should notice that the terms complementary type and C-MOS in this specification are used with broad meaning, and they also refer to the case in which a P-channel MOS and an N-channel MOS used in a ROM, for example, are respectively manufactured separately using both a P-channel MOS and an N-channel MOS when the entire circuit is viewed.

Brief description of the figures

Figure 1 is a block diagram, which indicates an application example of the micro-computer in this invention.

Figure 2 is a timing diagram, which shows the operational state in the application example illustrated in Figure 1.

Figure 3 is an electric circuitry diagram of a major circuit in the application example illustrated in Figure 1.

Figure 4 is a timing diagram, which shows the operational state of the circuit illustrated in Figure 3.

1...Micro-computer, 2...ROM, 3...RAM, 4...arithmetic circuit, 5...clock generating circuit, 6...instruction PLA, and 7...halt signal gate circuit.

Figure 1

Key:	A	Decoder
	B	Program counter stack
	C	Page stack
	D	R output latch
	E	Adder/comparator
	F	O output PLA
	G	O output latch
	5	System clock generating circuit
	6	Instruction PLA

Figure 2

Key:	1	Clock cycle
	2	ROM (complete dynamic logic)
	3	RAM (complete dynamic logic)
	4	ALU (random logic)
	5	Halt signal
	6	Nth command fetch cycle
	7	N-1 st command execution cycle
	8	ROM address output
	9	Fetch
	10	Command decode
	11	PC renewal
	12	Execution of branch/call
	13	Nth command execution cycle
	14	N-1 st command fetch cycle
	15	Command execution
	16	Reading of the RAM data
	17	Writing into RAM
	18	ALU input
	19	Storing into the register

20	K input operation
21	K input is effective
22	Pre-charge
23	Preservation of the information is possible
24	Halt accepting period
25	Halt operation
26	Time

Figure 3

Figure 4

Key: 1 Clock cycle
2 Halting operation
3 Time